BLACK FRIDAY SALES PREDICTION

Project Introduction

Black Friday is an informal name for the Friday following Thanksgiving Day in the United States, celebrated on the fourth Thursday of November. The day after Thanksgiving has been regarded as the beginning of the United States Christmas shopping season since 1952, although the term "Black Friday" did not become widely used until more recent decades. Many stores offer highly promoted sales on Black Friday and open very early, such as at midnight, or may even start their sales at some time on Thanksgiving. The major challenge for a retail store or eCommerce business is to choose product prices such that they get maximum profit at the end of the sales. This project deals with determining the product prices based on historical retail store sales data. After generating the predictions, our model will help the retail store decide the price of the products to earn more profits.

Dataset Description

The dataset is acquired from an online data analytics hackathon hosted by Analytics Vidhya. It contains features such as age, gender, marital status, categories of products purchased, city demographics, purchase amount, etc. The data consists of 12 columns and 537,577 records. Our model will predict the purchase amount of the products.

Exploratory Data Analysis (EDA)

Observations made during EDA:

- Approximately 75% of purchases are made by male users, indicating that male consumers are major contributors to the number of sales.
- Single men tend to spend the most during Black Friday, while men tend to spend less once they are married.
- Consumers aged 25-40 tend to spend the most.
- People who have spent 1 year in the city tend to spend the most.
- City B contributes significantly to the overall sales income, but City C is majorly responsible for the sales of specific products.

Data Preprocessing

- **Encoding categorical variables**: LabelEncoder was used for columns like Age, Gender, and City_Category.
- **Converting Stay_In_Current_City_Years**: This column was converted into numerical values.
- **Filling missing values**: Missing values in Product_Category_2 and Product_Category_3 were filled with their mean values.

```
# Importing necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor

# Load the dataset
data = pd.read_csv("BlackFridaySales.csv")

# Data preprocessing
# Encoding categorical variables
label_encoders = {}
categorical_cols = ['Gender', 'Age', 'City_Category']
for col in categorical_cols:
    label_encoders[col] = LabelEncoder()
    data[col] = label_encoders[col].fit_transform(data[col])

# Convert Stay_In_Current_City_Years to numerical values
data['Stay_In_Current_City_Years'] =
data['Stay_In_Current_City_Years'].str.extract('(\d+)')
data['Stay_In_Current_City_Years'] =
data['Stay_In_Current_City_Years'].astype(float)

# Filling missing values
data['Product_Category_2'] =
data['Product_Category_2'].fillna(data['Product_Category_2'].mean())
data['Product_Category_3'] =
data['Product_Category_3'].fillna(data['Product_Category_3'].mean())
```

Modeling Phase

- The dataset was split into random train and test subsets with a ratio of 80:20.
- Supervised learning models such as Linear Regression, Decision Tree Regressor, Random Forest Regressor, and XGBoost Regressor were implemented.

```
# Splitting the data into train and test sets
X = data.drop(columns=['Purchase', 'User_ID', 'Product_ID'])  # Remove non-
predictive columns
y = data['Purchase']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize models
models = {
    "Linear Regression": LinearRegression(),
    "Decision Tree Regressor": DecisionTreeRegressor(),
    "Random Forest Regressor": RandomForestRegressor(),
    "XGBoost Regressor": XGBRegressor()
}

# Train and evaluate models
results = {}
```

```
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    rmse = mean_squared_error(y_test, y_pred, squared=False)
    results[name] = rmse
```

Evaluation Metric

Root Mean Square Error (RMSE) is a standard way to measure the error of the models in predicting quantitative data. It is the square root of the average of squared differences between prediction and actual observation.

```
# Find the best performing model
best_model = min(results, key=results.get)
best_rmse = results[best_model]

# Print evaluation results
print("Evaluation Results:")
for name, rmse in results.items():
    print(f"{name}: RMSE = {rmse}")

print(f"\nBest Model: {best_model} (RMSE = {best_rmse})")
```

Conclusion

Multiple supervised models were implemented, and based on the RMSE scores, the XGBoost Regressor was the best performer with an RMSE score of 2879.

This documentation provides an overview of the Black Friday Sales Prediction project, including its objectives, dataset description, exploratory data analysis, data preprocessing steps, modeling phase, evaluation metric, and conclusion. Additionally, sample code snippets are provided for reference.